

REMARKS

Claims 1 - 21 are pending in the present patent application. The Examiner has rejected claims 1 - 21. Applicant has amended claims 1, 11, 18 and 21 and added new claim 22. Applicant respectfully requests reconsideration of claims 1 - 21 and consideration of claim 22 in view of at least the following amendments and remarks.

I. Rejection of Claims 1 - 21 Based on 35 U.S.C. § 102(e)

The Examiner has rejected claims 1 - 3, 15, 16, 18 and 19 under 35 USC §102(e) as being anticipated by Crutcher (U.S. Patent No. 5,844,560). The Examiner states:

3. Claims 1-3, 15, 16, 18, 19 are rejected under 35 U.S.C. 102(e) as being anticipated by Crutcher et al (5844560).

4. Regarding claims 1-3, 15-16, 18-19, see in Crutcher et al: the abstract, Figures 3, 5, 7, column 2 lines 1-19 and 59-68, column 3 lines 18-35 and 54-68 (note the change in the element when input device is detected and when therefore its handling code is associated with the element), column 4 lines 15-38 (again note how the element's look is modified), column 7 lines 48-68. Note that these claims are broad and recite merely that the runtime version of the element is examined and subsequently identified as supporting the input device. This is status' indication of the input device, and the element is marked or modified accordingly. This is shown in the aforesaid, with the computer system examining the element at runtime and determining whether the input device is affecting it (which would imply that the device's handling code is associated with it).

Applicant respectfully disagrees and submits that claims 1 - 22 are allowable for at least the following reasons:

1. Crutcher does not examine a runtime version of a screen element of a graphical user interface to detect an ability to process an input device's events and automatically identify the screen element as supporting the input device when input device-handling program code is associated with the screen element.

Applicant respectfully submits that Crutcher does not examine a runtime version of a screen element of a graphical user interface (GUI) to detect an ability to process an input device's events and automatically identify the screen element as supporting the input device when input device-handling program code is associated with the screen element. Crutcher is concerned with a graphical user interface that indicates whether or not it, and its associated function, have been activated. The invention consists of an image of a rocker button on a computer screen that grows smaller on one end when it is depressed using a mouse and larger on the opposite end, indicating that the larger end that is not depressed and has moved upward. The rocker button status and position must be continuously monitored by a computer program to modify the associated function being controlled by the rocker button. Crutcher is not attempting to determine if computer code is present in a program *before* it runs with the goal of

determining if support exists for an input device. The invention of Crutcher is not attempting to locate such support or determine if it exists at all, it is attempting to control the already existing support and indicate whether the device being controlled has been activated by means of changing the appearance of the rocker button. These are entirely different problems. Crutcher does not examine a runtime version of a screen element of a graphical user interface to detect an ability to process an input device's events and automatically identify the screen element as supporting the input device when input device-handling program code is associated with the screen element. Applicant therefore respectfully submits that the prior art in Crutcher does not teach, suggest or describe the claimed invention.

II. Rejection of Claims 4 - 14, 17 and 19 - 21 Based on 35 U.S.C. §103

The Examiner has rejected claims 4 - 14, 17 and 19 - 21 under 35 USC §103(a) as being unpatentable over Crutcher (U.S. Patent No. 5,844,560) in view of Carey et. al (U.S. Patent No. 6,122,627). The Examiner states:

6. Claims 4-14, 17, 19-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Crutcher et al (5844560) in view of Carey et al (6122627).

7. Regarding claim 4, in addition to the aforementioned, Crutcher et al do not go into the details of examining during the construction of an element, but do mention flexibility in examining the element. Also, see in Carey et al: the Abstract, Figure 7, Figure 9A, column 3 lines 58-68, column 4 lines 1-10, column 5 lines 34-60, column 9 lines 47-60, column 10 lines 5-26, column 21 lines 1-33 for example. This shows how an element is examined during construction. It would have been obvious to a person with ordinary skill in the art to do this in Crutcher

et al as well, for added flexibility in a system that examines elements.

8. Regarding claims 5-8, 10-13, 15-17, 19-20, Crutcher et al may not go into the details of the class definitions, superclasses, and interface declarations, but these are properties that are associated with interface elements. This is shown in Carey et al: see also column 6 lines 10-52, column 7 lines 32-68, column 8 lines 1-20, column 11 lines 16-68, column 15 lines 18-53, column 16 lines 47-68. It would have been obvious to a person with ordinary skill in the art to have this in Crutcher et al because it would provide a convenient way to use the elements

9. Regarding claims 9, 14, 18, 21, Crutcher et al may not go into the details of whether the element delegates the processing of the input to other code, but do show flexibility in handling elements, and Carey et al show delegating various element processes. Delegating to other code is common in the art as a flexibility for handling elements. It would have been obvious to a person with ordinary skill in the art to do this in Crutcher et al because it would provide a convenient way to add flexibility to element handling.

Applicant respectfully disagrees and submits that claims 4 - 14, 17 and 19 - 21 are allowable for at least the following reasons:

1. Crutcher and Carey, either alone or in any combination of the two, do not examine a runtime version of a screen element of a graphical user interface to detect an ability to process an input device's events and automatically identify the screen element as supporting the input device when input device-handling program code is associated with the screen element.

Applicant respectfully submits that Crutcher and Carey, either alone or in any combination of the two, do not examine a runtime version of a screen element of a graphical user interface to detect an ability to process an input device's events and automatically identify the screen element as supporting the input device when input device-handling program code is associated with the

screen element. In the preceding section, Applicant has already discussed why Crutcher does not teach, describe or suggest the claimed invention. Indeed, the Examiner has conceded that "Crutcher et al do not go into the details of examining during the construction of an element".

Applicant respectfully submits that Carey does not examine a runtime version of a screen element of a graphical user interface to detect an ability to process an input device's events and automatically identify the screen element as supporting the input device when input device-handling program code is associated with the screen element. Carey is concerned with developing a tool to be used to obtain different specific "views" of data in a relational database management system. Upon receipt of a query referencing a view type, a query engine generates a query plan that builds mock, or proxy, application type objects in computer memory that do not contain data. An application can run methods on the application objects. The proxy objects can redo the same calculations that a view would do to obtain data. In this way, Carey builds objects from view definitions and queries. Carey does not examine a runtime version of a screen element of a graphical user interface to detect an ability to process an input device's events and automatically identify the screen element as supporting the input device when input device-handling program code is associated with the screen element. Carey works with a database that is known to exist and looks for views (or portions) of data that are known to exist. Carey

has the capability to construct objects using these views. Carey is not concerned with examination of a runtime version of a screen element of a graphical user interface to detect an ability to process an input device's events and automatically identify the screen element as supporting the input device when input device-handling program code is associated with said screen element. Carey does not have anything remotely similar to this capability because, in Carey, the existence of the appropriate data in the database is unquestioned because it is known to exist. Accordingly, Applicant submits that Carey does not examine a runtime version of a screen element of a graphical user interface to detect an ability to process an input device's events and automatically identify the screen element as supporting the input device when input device-handling program code is associated with the screen element. Applicant therefore respectfully submits that the prior art in Carey does not teach, suggest or describe the claimed invention.

Applicant respectfully submits that Crutcher and Carey, either alone or in any combination of the two, do not examine a runtime version of a screen element of a graphical user interface to detect an ability to process an input device's events and automatically identify the screen element as supporting the input device when input device-handling program code is associated with the screen element. The invention of Crutcher is attempting to control the already existing support for a screen element and indicate whether the device being

controlled has been activated by means of changing the appearance of the rocker button. Carey is concerned with building objects from database view definitions and queries. If these two inventions were combined as proposed by the Examiner, the combination of the two suggests attempting to control the building of objects from database views and determining if they have been activated. This combination does not suggest anything remotely resembling the claimed invention. Crutcher and Carey, either alone or in any combination of the two, do not examine a runtime version of a screen element of a graphical user interface to detect an ability to process an input device's events and automatically identify the screen element as supporting the input device when input device-handling program code is associated with the screen element.

The MPEP states at Section §2143, page 2100 - 97:

To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations.

Applicant respectfully submits that a *prima facie* case of obviousness has not been established. The Examiner has proposed to combine the prior art in Crutcher and Carey because it would have been obvious to a person of ordinary skill in the art to do this.

While combining the prior art in the Crutcher and Carey may improve either one, even in combination, the prior art in Crutcher and Carey do not teach the claimed invention. There is no suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the references, or to combine reference teachings, to generate the invention claimed here. Applicant submits that the procedures discussed in Crutcher are attempting to control the already existing support for a screen element and indicate whether the device being controlled has been activated by means of changing the appearance of the rocker button. Carey is concerned with building objects from database view definitions and queries. These goals are quite different from the claimed invention and Applicant contends that there is no motivation in the prior art in Crutcher or Carey to combine the reference teachings. As discussed in the previous paragraphs, a combination of Crutcher and Carey does not suggest the claimed invention. Applicant respectfully submits that Crutcher and Carey, either alone or in any combination of the two, do not examine a runtime version of a screen element of a graphical user interface to detect an ability to process an input device's events and automatically identify the screen element as supporting the input device when input device-handling program code is associated with the screen element. Applicant therefore respectfully submits that the prior art in Crutcher and Carey does not teach, suggest or describe the claimed invention.

III. Remarks Regarding Paper Number 13

In the Office Action, dated March 13th, 2001 (paper number 13), the Examiner rejected claims 1-21 under 35 USC 103(c) stating the claim invention was obvious based on the Ford reference in view of the Rafecz reference. Applicant withdraws the arguments submitted on June 12th, 2001 and submits new arguments relating to the Ford and Rafecz references. Applicant respectfully requests that the Examiner reconsider the applicability of the Ford and Rafecz references in view of 35 USC 102(a) and consider the following new arguments.

The Examiner has rejected claims 1 – 21 in the March 13, 2001 office action under 35 USC §103(a) as being unpatentable over Ford (U.S. Patent No. 5786815) in view of Rafecz et. Al (U.S. Patent No. 5940494). The Examiner states:

Regarding claims 1-21, see Ford: Abstract, Figures 1B, 8A, column 2 lines 16-48, column 3 lines 20-62 (note the GUI receiving input file data), column 4 lines 5-50 (note the widget elements, the nesting of data classes and class definitions), column 5 lines 1-40 and column 15 lines 5-36 (note the widget indicia that support the input data and modification of the screen widget elements). Ford may not specifically describe the showing the detection of the input device when identifying associated program source code, or thus showing the specific updating of the input device status, but he does show the display and modifications of GUI elements that support input source code. Furthermore, see Rafecz et al: Abstract, Figure 2, column 2 lines 44-60, column 45 lines 19-40 for example. This shows updating elements based on detecting input device data, in which the updating is reflected in a graphical change in the element. It would have been obvious to a person with ordinary skill in the art to incorporate this possible feature of modifying the element for input device detection, into the system of Ford, because it would provide an efficient way to display and modify GUI elements that support input source code.

Applicant's arguments filed have been fully considered but they are not persuasive. The citations above show the relevant portions of the patent. For example, see column 5 lines 10-22 in Ford. These show updating the GUI elements based on modifications in the input data file, which in turn are due to input device changes. The crux of the argument lies in the interpretation of applicants' claims. Note they are broad and any GUI that may change elements due to input device changes is relevant art. Note also that detecting and identifying screen elements,

in view of the application, may be interpreted the same way. Applicant is invited to contact examiner to discuss claim language.

Applicant believes the claimed invention is not anticipated by the Ford and Rafecz references for at least the following reasons.

1. Ford and Rafecz, either alone or in any combination of the two, do not examine a runtime version of a screen element of a graphical user interface to detect an ability to process an input device's events and automatically identify the screen element as supporting the input device when input device-handling program code is associated with the screen elements.

Applicant respectfully submits that Ford and Rafecz, either alone or in any combination of the two, do not examine a runtime version of a screen element of a graphical user interface to detect an ability to process an input device's events and automatically identify the screen element as supporting the input device when input device-handling program code is associated with the screen element.

Applicant respectfully submits that Ford does not examine a runtime version of a screen element of a graphical user interface to detect an ability to process an input device's events and automatically identify the screen element as supporting the input device when input device-handling program code is associated with the screen element. Ford is concerned with modification and specification of a graphical user interface without a need to specify callback routines and form a new application executable. Thus, user driven graphical user interface changes can be achieved without user access to the application source

code or the application developer's involvement. Ford is not attempting to determine if computer code is present in a program before it runs with the goal of determining if support exists for an input device. The invention of Ford is not attempting to locate such support or determine if it exists at all; it is attempting to modify a graphical user interface without a need to for access to the application source code. These are entirely different problems. Ford does not examine a runtime version of a screen element of a graphical user interface to detect an ability to process an input device's events and automatically identify the screen element as supporting the input device when input device-handling program code is associated with the screen element. Applicant therefore respectfully submits that the prior art in Ford does not teach, suggest or describe the claimed invention.

Applicant submits that Rafecz also does not teach detecting input device support of a screen element of a graphical user interface or identifying the screen element as supporting an input device when an input device-handling program code is associated with the element. Applicant submits that Rafecz discusses a method for displaying real time data relating to the operation of an automatic call distributor where an *agent* (a person) has the capability from an agent display to modify the types of data displayed, how it is displayed and the time period for data updates. The agent has to manually make the changes. Applicant believes that the method of Rafecz does not detect input devices. Therefore, Applicant submits that Rafecz also does not teach detecting input device support of a screen element of a graphical user interface or identifying the screen element as

supporting an input device when an input device-handling program code is associated with the element.

Applicant respectfully submits that nothing in the portions of the Rafecz patent cited by the Examiner, nor in any combination of the Ford and Rafecz patents, describes *detecting* input device support of a screen element of a graphical user interface or *identifying* the screen element as supporting an input device when an input device-handling program code is associated with the element. If the Ford and Rafecz inventions were combined, the result would be an invention for modifying a graphical user interface, without access to source code, for a telephone call distributor operated by a person with the capability to allow the person to modify the data display. Ford and Rafecz do not teach or suggest a method of detecting input device support of a screen element of a graphical user interface or identifying the screen element as supporting an input device when input device-handling program code is associated with the screen element as claimed in claims 1, 10, and 15. Claim 18 describes "a detector configured to examine a runtime version of said screen element to identify whether said screen element supports said input device by determining whether input device-handling program code is associated with said screen element".

Applicant respectfully submits that nothing in the portions of the Ford and Rafecz patents cited by the Examiner describe a *detector* configured to examine a runtime version of a screen element to identify whether the screen element supports the input device by determining whether input device-handling program code is associated with the screen element.

The MPEP states at Section §2143, page 2100 – 97:

To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations.

Applicant respectfully submits that a *prima facie* case of obviousness has not been established. The Examiner has proposed to combine the prior art in Ford and Rafecz because it would have been obvious to a person of ordinary skill in the art to do this.

While combining the prior art in the Ford and Rafecz may improve either one, even in combination, the prior art in Ford and Rafecz do not teach the claimed invention. There is no suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the references, or to combine reference teachings, to generate the invention claimed here. Applicant submits that the procedures discussed in Ford are aimed at allowing users to change graphical user interfaces without access to application source code. Rafecz is concerned with an agent (a person) making changes in data displays in a piece of telephone equipment. The combination of the art in Ford and Rafecz, as discussed earlier, would yield an invention for modifying a graphical user interface, without access to source code, for a telephone call distributor operated by a person with the capability to allow the person to modify the data display. The combination of Ford and Rafecz yields

an invention with goals quite different from the claimed invention and Applicant respectfully submits that Ford and Rafecz, either alone or in any combination of the two, do not examine a runtime version of a screen element of a graphical user interface to detect an ability to process an input device's events and automatically identify the screen element as supporting the input device when input device-handling program code is associated with the screen element. Applicant therefore respectfully submits that the prior art in Ford and Rafecz does not teach, suggest or describe the claimed invention.

Dependent Claims 1 – 21

Applicant respectfully submits that claims 1 – 21 being dependent upon respective allowable base claims are also allowable for at least the foregoing reasons stated above.

CONCLUSION

For at least the foregoing reasons, Applicant respectfully submits that pending claims 1-21 and new claim 22 are patentably distinct from the prior art of record and in condition for allowance. Applicant therefore respectfully requests that pending claims 1-22 be placed in condition for allowance.

Respectfully submitted,

THE HECKER LAW GROUP

Date: 12/27/01

By: Obi I. Iloputaife
Obi I. Iloputaife
Reg. No. 45,677

THE HECKER LAW GROUP
1925 Century Park East
Suite 2300
Los Angeles, California 90067
(310) 286-0377

CERTIFICATE OF MAILING

This is to certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as First Class Mail in an envelope addressed to Assistant Commissioner for Patents, Washington, D.C. 20231 on December 27, 2001.

Christine Mills 12-27-01

Signature: Christine Mills Date: December 27, 2001